

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.

CENTRAL PROGRAMMING SYSTEM WITH DISTRIBUTED PLAYBACK DEVICES

RELATED APPLICATIONS

This application is based on and claims priority to U.S. Provisional Patent Application Nos. 60/219,764, filed July 20, 2000.

FIELD OF INVENTION

The invention is directed generally to the field of digital publishing and specifically to electronic playback devices such as TVs, CD players, Radios.

OVERVIEW

The playback devices (adaptors) (100) are made up of a computer, a display, optional speakers, optional infra-red transmitters, and means for connecting to the internet, whether that is a modem, DSL, T1, wireless or some other means.

The central program server (CPS) (200) accepts connections from clients (300). A user (1000) uses a client (300) to access the central program server using a standard userid/password combination. A user (220) is a logical entity that is recognized by the central program server (CPS) and has certain attributes such as a role, permissions, and quotas.

The CPS stores playback content (600) for all playback devices located physically in remote locations around the world. The CPS creates a playback script (5000) for each

playback device and delivers said playback script to said each playback device. The CPS can create a playback script for each playback device either in advance of or in response to a request by a playback device.

PLAYBACK DEVICE

FEATURES

The playback device is a hybrid analog/digital device. The Sidecast TV™ device, for example, can play digital playback content (aka Sidecast™) from the CPS while also showing live broadcast TV received through a cable-TV cable or an aerial antenna. A SidecastCD™ device can play digital playback content (aka Sidecast™) from the CPS while also playing an inserted music CD-ROM. Similarly, a SidecastDVD™ player with a movie DVD.

HARDWARE FEATURES

Modem to connect to the Internet using a dial-up connection from a national ISP at a cost of about \$19.95 or down to \$8.95 in bulk.

Disk drive of 2Gigabytes minimum to store the operating system and playback content.

CD-ROM drive or ZIP drive to allow for input of data at higher bandwidth than modem and for overriding action of retrieving playback content from Internet. For a SidecastCD™ device, a CD-ROM is also used to playback music CDs.

CPU, memory (64MB), optional sound card, and video chip.

TV card (110) that can tune in television signals and interface with the video display sub-system. (WinTV® is a brand that works well). This is used in the SidecastTV™ device. (optional)

A display screen (101). (optional, but necessary most of the time)

A DVD player (102). (optional)

An infra-red transmitter or IR port (104) to transmit data via an IR link. (optional)

Optional keyboard and mouse for manual set-up, maintenance, and overriding settings. (105)

Power supply that can be shutdown under software control. (optional) (106)

SOFTWARE FEATURES

Reboot (2500) the operating system. This, when done periodically, helps to keep the system fresh from memory leaks and other problems associated with a buggy operating system.

Windows ® Operating System (2800) is used because of the vast software tools and applications available.

A Display Engine (2000) which can show:

- HTML files (2100) along with all related file types such as Flash, Shockwave, GIF, JPEG, MPEG, text, scrolling text, and so on.
- TV image (2200) that it receives by interfacing with the TV card. The TV image (2200) can be shown in any rectangular area (masked by any other shape to achieve special effects) anywhere on the screen.

The display engine can show just the HTML file or the HTML file and the TV image.

The TV image is controlled by a special tag (2210) in the HTML file. Optionally the specification can be given in a file with the same name as the HTML file, but with a “.TV” extension instead of a “.HTML” or “.HTM” extension.

The TV specification gives:

- the X and Y coordinates of the top/left corner of the TV image as a percentage of the total screen dimensions or in pixel units and similarly gives the width and height of the TV image.
- The channel to tune the TV card.
- Any other accessible parameters of the TV card, such as saturation, hue, tint, volume, and so on.

A sample specification would be:

<TVSPEC T=0 L=0 W=480 H=450>

ACCESSING THE CPS AND PLAYBACK

A playback device accesses the CPS as a user with a userid/password combination and an identifier called the AdaptorId.

A playback device retrieves a compressed archive file (a .ZIP file) from the CPS. This file is uncompressed and its contents are stored on the local disk.

Playback content retrieval is scheduled to occur periodically and/or when the playback device is booted or turned on (for the case where the playback device is manufactured with a built-in or instant-on operating system).

When the playback device starts, it runs the display engine software. The display engine automatically reads a file called page1.html (this can be overridden by a command-line option). The initial page1.html file is created at start-up by copying a pre-

installed file called splashScreen.html to page1.html. SplashScreen.html shows a start-up screen on the display and periodically refreshes using JavaScript. The result is that when a content retrieval operation updates page1.html, the most recently created page1.html file replaces the splash screen.

Other start-up operations include the logging of the start time in a log file.

When content is retrieved from the CPS, the log file can be sent back to the CPS.

When content is retrieved from the CPS, the display engine software and other system software for the playback device can also be retrieved and installed. This allows for remote maintenance and updates.

PROGRAMMING THE CPS

The Central Programming Server is a web-server (201) with a database (202) and code (203) to:

- Create an account (210)
- Create a user (220)
- Create a magazine (500)
- Create a magazine issue (501)
- Create a magazine page (502)
- Create a magazine page layout (503)
- Create a magazine page layout area (504)
- Create a magazine page layout area loop-position (505)
- Receive and store playback content (600)

- Create a location record (700)
- Create an adaptor (playback device) record (800)
- Create a log record (230)
- Create playback instructions (5000) (e.g. HTML with JavaScript) for a playback device
- Associate playback content with a particular magazine page layout area loop-position, optionally with a time-dependency so that only at a particular time of day or date or day of the year, the given playback content is actually played.

The CPS is programmed for the following concepts:

- That a set of playback devices are installed in many physical locations.
- That the playback devices can be symbolically labeled and later retrieved individually or as groups, based on attributes. For example, a set of playback devices can be labeled with “Front Window” and others with “Cashier Area” and then programming can be applied to the set of all “Front Window” adaptors.
- That physical locations where playback devices (aka “adaptors”) are installed are owned by an owner, called a Host account (211).
- That the locations can be labeled in the same way as the playback devices.
- That the playback devices have hardware limitations and resources, such as the screen size and resolution of the physical screen connected to a given playback device, or that a playback device has IR capability.
- That the CPS administrators are super users and can access and change all data.

- That users belonging to a Host account (211) if given the role of editor or super user, can create magazine issues, manage the pages, layout, and content of said magazine issues and also control the distribution of said magazine issues to the playback devices. Playback devices to receive a magazine can be specified by zip-code, by location labels (720), by playback device label (820), or some other search criteria for playback devices.
- That there are entities that would like to provide content on a fee basis to be shown on a set of playback devices potentially across magazines and across Hosts. An Advertiser account (212) is for these entities.
- That users belonging to an Advertiser account (212) can load advertisements into the CPS that will be treated by the CPS as playback content, but also recognized as something for which a fee should be charged, i.e. an advertisement placement.
- That screen-time on playback devices as well as IR transmission time can be sold to Advertisers. Advertisers have accounts for billing and e-commerce transactions of a B-to-B nature will be conducted between the CPS owning entity (Ch1 Communications Inc., for example) and the Advertising entity.
- That advertising, if not all playback content, will be measured in duration in some units, for example, seconds or 7.5-second units, and that fees will be charged for the use of screen-space and for the use of screen-time. The fees will be calculated based on a formula taking into consideration the size of the magazine page layout area and the duration of the magazine page layout area loop-position as well as special specifications, such as time-of-day, day-of-month, day-of-year, and so on.

- That each Advertiser user (222) can only view the advertisement playback content (670) owned by that advertiser.
- That a Host user (221) can designate any number of magazine page layout area loop-positions to be available for advertisers.
- That a Host user (221) can designate any number of magazine page layout area loop-positions to be available to another Host user by specifying that user explicitly, by label, by account, by predefined category of users, where the user members of that category of users can be specified by listing explicitly the users, by user labels, by accounts, and so on.
- That accounts and users can have associated labels in order that groups of like or unlike accounts and users can be created.
- That playback devices will periodically access the CPS using a userid/password and adaptorId to identify themselves and request the latest playback content.
- That a magazine page layout area can be filled with content that is represented or specified using a data file, such as an image, a video, an audio file, or text.
- That a magazine page layout area can be filled with auto-generated content such as time, temperature reading, TV signal, AM/FM radio signal, or other sensor-generated or playback device-generated information at the time of playback by the playback device.
- That a magazine page layout area can be automatically filled by the CPS using random or otherwise loosely specified content from a playback content category.

The CPS presents a set of web-pages for interaction with a user to manage accounts, users, magazines, playback content, advertising playback content, billing, viewing

playback device performance, viewing schedules and show statistics of playback devices, managing groups or categories of accounts, users, playback devices, locations, and playback content. The basic grouping or categorizing mechanism is the application of a text label to the corresponding database record. Any number of labels can be applied. The corresponding records can be retrieved using an "AND" or "OR" of labels.

The CPS can create an HTML file with the appropriate JavaScript code for playback of a magazine issue or a single page of a magazine issue, or a single layout area of a magazine page, or a single loop-position of a single layout area of a magazine page. Using a checkbox, a user can check each layout area and loop-position to be viewed. If the user has sufficient viewing privileges for a given playback content, then that playback content is shown, otherwise, a place-holder image is shown.

Where the playback device has an embedded or local web server, the playback engine retrieves information locally from the local web server. The local web server retrieves content from the CPS. The local web server (WEBSERVERNUMBER) can interpret custom markup tags in the playback information and perform local actions. One example, is a playback device with a TV tuner, wherein the playback display engine retrieves a page of information to display and the local web server in addition to supplying the information to the display engine interprets specifications for the position and channel of a TV picture, tunes the TV tuner and arranges for the TV picture to be displayed, either superimposed or under the display of the display engine output.

All claimed elements are software elements, methods, and systems which coordinate the delivery, reception, interpretation, and physical interfaces of data.

ADVANTAGES

This publishing system allows various and many content providers to provide content for a particular addressable playback device or group of playback devices, whether one or 10,000 devices, whether down the street, or around the world.

Content is categorized by each user.

Content is delivered on a timely basis efficiently and effectively using the Internet.

Content is secure on the CPS and on each playback device.

Content providers, including advertisers especially, can get a report of the times and places their content was shown.

IR data including coupons, directions, or other information can be beamed to PDA devices or other suitable receiving devices in proximity to a playback device.

Local playback devices, with the local web server architecture described or a similar one, can locally interpret and produce augmented and specific results. For example, showing the local time or the local temperature, or the live local output of an attached video camera, VCR, DVD, or other device.

ADVERTISEMENT PLACEMENT

A user of an Advertiser account can upload digital advertisement content.

SPECIAL CONSIDERATIONS

The CPS is designed with the following optimizations and special considerations:

- A playback content is stored only once and referenced by each magazine page layout area loop-position record that references it.
- Playback content is delivered to each playback device once and is not transmitted again unless the playback content is changed since it was delivered to the playback device or the playback content is removed from the playback device.

- A new magazine issue can be copied from a previous magazine issue in order to save editing time.
- Pages of a magazine issue can be copied to a clipboard and pasted into a magazine issue.
- Layout areas of a magazine issue can be copied to a clipboard and pasted into a magazine issue. Ideally, the magazine page layout areas should be compatible in size or the playback content should not be sensitive to magazine page layout area size.

EXHIBITS

- 1) Diagram of Entire System
- 2) Diagram of CPS components
- 3) Diagram of SidecastTV
- 4) Printout of Database Schema
- 5) Printout of sample playback device script with TV tag.
- 6) Printout of PHP code modules.
- 7) Screen shots of Sidecast Engine (Web-site)
- 8) Printout of the “cookbook”

PLAYBACK DEVICE

The Playback Device is a display engine. It functions as a means for showing multimedia content through a browser. It is capable of showing and looping multimedia content at specific times, including, but not limited to, daily specific periods. It also logs the start and stop times of the multimedia loop. The Playback Device is able to accept CDs, zip drives and can load information from the Internet or local disks. The current embodiment of the System utilizes a modem and other similar means of **communication (?)**. Admittedly, there are security issues that are unavoidable but which are best resolved with the use of the modem.

The display engine consists of two components: the television area and the Sidecast area. The areas are adjacent to each other. This split-screen television contains a layout interpreter. The layout interpreter is capable of interpreting a layout specification. The layout specification is an HTML web-page. The web-page contains at least one tag. The layout interpreter interprets the tag. In addition, the layout interpreter also executes the following tasks: reading a layout specification file, positioning a TV area, sizing a TV area and setting the channel or other properties of the TV area. Setting the channel is a determination to be made by the host of the device and is subject to change as often as desired.

The display engine is also a means for transmitting and receiving IR data and is capable of coordinating IR data with multimedia shown on the screen.

PUBLISHING SYSTEM



The Publishing System is an online publishing system that consists of four (4) components: a client computer, a server computer, a second client computer, and a database. The Publishing System creates a plurality of magazine records and magazine page records in said database. The magazine records contain different versions and combinations of screen layouts. The user may edit or view the layouts acting in the capacity of host, manager, editor, etc. The layouts contain three areas of content. For any given area of content, the user may elect to choose content from the content gallery. While choosing the content gallery, user refers to a thumbnail picture of the layout. The user can choose and control the timing of the content selection. The user can create at least one layout area where playback is automatic. The system has the potential to store content.

The system can create an HTML file with embodied specifications for TV operation. The system has the means for creating HTML instructions representing a page. The system may also emit and receive IR signals that may be stored in playback device for delivery to said publishing system at either a later time, a pre-defined point in time or periodically.

[illegible]

Julia

MySQL dump 6.0

#

Host: localhost Database: tsn

#-----

Server version 3.22.25

#

Table structure for table 'Adaptor'

#

DROP TABLE IF EXISTS Adaptor;

CREATE TABLE Adaptor (

id bigint(21) DEFAULT '0' NOT NULL auto_increment,

location_id bigint(21) DEFAULT '0' NOT NULL,

account_id bigint(21) DEFAULT '0' NOT NULL,

note varchar(255),

PRIMARY KEY (id)

);

#

Table structure for table 'AdaptorLocation'

#

DROP TABLE IF EXISTS AdaptorLocation;

CREATE TABLE AdaptorLocation (

id bigint(21) DEFAULT '0' NOT NULL auto_increment,

address varchar(255) DEFAULT '' NOT NULL,

city varchar(64) DEFAULT '' NOT NULL,

state varchar(64) DEFAULT '' NOT NULL,

zip varchar(9) DEFAULT '' NOT NULL,

note varchar(255) DEFAULT '' NOT NULL,

account_id bigint(20) DEFAULT '0' NOT NULL,

hours_on smallint(6) DEFAULT '8' NOT NULL,

PRIMARY KEY (id)

);

#

#

Table structure for table 'AdaptorLog'

#

DROP TABLE IF EXISTS AdaptorLog;

CREATE TABLE AdaptorLog (

adaptor_id bigint(21) DEFAULT '0' NOT NULL,

timestamp timestamp(14),

event varchar(255) DEFAULT '' NOT NULL

);

#

Table structure for table 'AdaptorScreen'

#

DROP TABLE IF EXISTS AdaptorScreen;

CREATE TABLE AdaptorScreen (

adaptor_id bigint(21) DEFAULT '0' NOT NULL,

screen_id bigint(21) DEFAULT '0' NOT NULL,

PRIMARY KEY (adaptor_id,screen_id)

);

#

Table structure for table 'AreaPermission'

#

DROP TABLE IF EXISTS AreaPermission;

CREATE TABLE AreaPermission (

user_id bigint(21) DEFAULT '0' NOT NULL,

permission varchar(64) DEFAULT '' NOT NULL,

PRIMARY KEY (user_id,permission)

);

#

Table structure for table 'Content'

```

...#
DROP TABLE IF EXISTS Content;
CREATE TABLE Content (
    id bigint(21) DEFAULT '0' NOT NULL auto_increment,
    content_file varchar(255),
    content_type varchar(64),
    duration int(11),
    valid_from datetime,
    valid_to datetime,
    name varchar(64),
    description varchar(255),
    keywords varchar(255),
    mod_date timestamp(14),
    user_id bigint(20) DEFAULT '0' NOT NULL,
    account_id bigint(20) DEFAULT '0' NOT NULL,
    size int(11),
    PRIMARY KEY (id)
);

#
# Table structure for table 'ContentCategory'
#
DROP TABLE IF EXISTS ContentCategory;
CREATE TABLE ContentCategory (
    id int(11) DEFAULT '0' NOT NULL auto_increment,
    category varchar(30),
    type varchar(32),
    PRIMARY KEY (id)
);

#
# Table structure for table 'ContentInCategory'
#
DROP TABLE IF EXISTS ContentInCategory;
CREATE TABLE ContentInCategory (
    content_id bigint(21) DEFAULT '0' NOT NULL,
    category_id bigint(21) DEFAULT '0' NOT NULL,
    detail varchar(64),
    PRIMARY KEY (content_id,category_id)
);

#
# Table structure for table 'LayoutTemplate'
#
DROP TABLE IF EXISTS LayoutTemplate;
CREATE TABLE LayoutTemplate (
    id mediumint(9) DEFAULT '0' NOT NULL auto_increment,
    name varchar(64) DEFAULT '' NOT NULL,
    image_file varchar(255) DEFAULT '' NOT NULL,
    notes varchar(255),
    PRIMARY KEY (id)
);

#
# Table structure for table 'LayoutTemplateArea'
#
DROP TABLE IF EXISTS LayoutTemplateArea;
CREATE TABLE LayoutTemplateArea (
    area_id tinyint(4) DEFAULT '0' NOT NULL,
    layout_id int(11) DEFAULT '0' NOT NULL,
    color varchar(32) DEFAULT 'green,#00FF00' NOT NULL,
    geometry varchar(64) DEFAULT '' NOT NULL,
    valid_content_types varchar(255) DEFAULT '' NOT NULL,
    PRIMARY KEY (layout_id,area_id)
);

```



```

#
# Table structure for table 'Magazine'
#
DROP TABLE IF EXISTS Magazine;
CREATE TABLE Magazine (
  id int(11) DEFAULT '0' NOT NULL auto_increment,
  name varchar(50),
  account_id bigint(21) DEFAULT '0' NOT NULL,
  new_issue_interval int(11),
  PRIMARY KEY (id)
);

#
# Table structure for table 'MagazineContent'
#
DROP TABLE IF EXISTS MagazineContent;
CREATE TABLE MagazineContent (
  page_id bigint(21) DEFAULT '0' NOT NULL,
  content_id bigint(21) DEFAULT '0' NOT NULL,
  duration int(11),
  area_id tinyint(4) DEFAULT '0' NOT NULL,
  transition smallint(6) DEFAULT '0' NOT NULL,
  transition_duration smallint(6) DEFAULT '0' NOT NULL,
  id bigint(21) DEFAULT '0' NOT NULL auto_increment,
  position mediumint(9) DEFAULT '0' NOT NULL,
  source int(11) DEFAULT '0' NOT NULL,
  status varchar(32),
  mod_date timestamp(14),
  PRIMARY KEY (id)
);

#
# Table structure for table 'MagazineIssue'
#
DROP TABLE IF EXISTS MagazineIssue;
CREATE TABLE MagazineIssue (
  magazine_id int(11) DEFAULT '0' NOT NULL,
  issue_id bigint(21) DEFAULT '0' NOT NULL auto_increment,
  valid_from datetime DEFAULT '0000-00-00 00:00:00' NOT NULL,
  PRIMARY KEY (issue_id)
);

#
# Table structure for table 'MagazinePage'
#
DROP TABLE IF EXISTS MagazinePage;
CREATE TABLE MagazinePage (
  issue_id bigint(21) DEFAULT '0' NOT NULL,
  page_id bigint(21) DEFAULT '0' NOT NULL auto_increment,
  layout_id mediumint(9) DEFAULT '0' NOT NULL,
  page_number mediumint(9) DEFAULT '0' NOT NULL,
  next_page mediumint(9) DEFAULT '0' NOT NULL,
  duration int(11) DEFAULT '0' NOT NULL,
  mod_date timestamp(14),
  PRIMARY KEY (page_id, issue_id)
);

#
# Table structure for table 'Screen'
#
DROP TABLE IF EXISTS Screen;
CREATE TABLE Screen (
  screen_id bigint(21) DEFAULT '0' NOT NULL auto_increment,
  manufacturer varchar(255) DEFAULT '' NOT NULL,
  model varchar(255) DEFAULT '' NOT NULL,
  aspect_ratio varchar(16) DEFAULT '16:9' NOT NULL,

```

```

résolution varchar(16) DEFAULT '1024x1024' NOT NULL,
notes varchar(255) DEFAULT '' NOT NULL,
PRIMARY KEY (screen_id);
);

#
# Table structure for table 'TSNAccount'
#
DROP TABLE IF EXISTS TSNAccount;
CREATE TABLE TSNAccount (
  id bigint(21) DEFAULT '0' NOT NULL auto_increment,
  name varchar(64) DEFAULT '' NOT NULL,
  contact_id bigint(21) DEFAULT '0' NOT NULL,
  type varchar(16) DEFAULT '' NOT NULL,
  PRIMARY KEY (id)
);

#
# Table structure for table 'TSNContact'
#
DROP TABLE IF EXISTS TSNContact;
CREATE TABLE TSNContact (
  id int(15) DEFAULT '0' NOT NULL auto_increment,
  salutation varchar(4),
  first_name varchar(15),
  middle_name varchar(15),
  last_name varchar(15),
  address_1 varchar(100),
  address_2 varchar(100),
  city varchar(30),
  state varchar(15),
  zipcode varchar(10),
  company varchar(30),
  title varchar(30),
  work_address_1 varchar(100),
  work_address_2 varchar(100),
  work_city varchar(30),
  work_state varchar(15),
  work_zipcode varchar(10),
  home_phone varchar(14),
  work_phone varchar(14),
  email varchar(30),
  note mediumtext,
  PRIMARY KEY (id)
);

#
# Table structure for table 'TSNSession'
#
DROP TABLE IF EXISTS TSNSession;
CREATE TABLE TSNSession (
  user_id bigint(21),
  last_in timestamp(14),
  last_out datetime,
  session_id varchar(64) DEFAULT '' NOT NULL,
  session_code text
);

#
# Table structure for table 'TSNUser'
#
DROP TABLE IF EXISTS TSNUser;
CREATE TABLE TSNUser (
  id bigint(21) DEFAULT '0' NOT NULL auto_increment,
  username varchar(30) DEFAULT '' NOT NULL,
  password varchar(30) DEFAULT '' NOT NULL,

```


1111 (double quotes)
are really newlines.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 FINAL//EN">
<HTML>
<HEAD>
<TITLE>Sidecast TV</TITLE>
<SCRIPT language="JavaScript">
function Start() {
document.body.scroll = 'no';
window.setTimeout('TransAreaContent1(
,14000);
window.setTimeout('TransAreaContent1(,17000);'
);
function
TransAreaContent1() {
Areal.filters.item(0).Apply();
Areal.filters.item(0).Transition = 3;
ArealContent1.style.visibility =
'hidden';
ArealContent1.style.visibility = 'visible';
document.FlashArealContent1
.rewind();
document.FlashArealContent1.play();
Areal.filters.item(0).play(1);
window.setTimeout('TransAreaContent2(,14000);'
);
function TransArea
Content2() {
Areal.filters.item(0).Apply();
Areal.filters.item(0).Transi
tion = 2;
ArealContent1.style.visibility = 'hidden';
ArealContent2.style.visibility
= 'visible';
document.FlashArealContent2.rewind();
document.FlashArealContent2.play(
);
Areal.filters.item(0).play(1);
window.setTimeout('TransAreaContent3(,
13000);'
);
function TransAreaContent3() {
Areal.filters.item(0).Apply
();
Areal.filters.item(0).Transition = 2;
ArealContent2.style.visibility =
'hidden';
ArealContent3.style.visibility = 'visible';
document.FlashArealContent3
.rewind();
document.FlashArealContent3.play(1);
Areal.filters.item(0).play(1);
window.setTimeout('TransAreaContent4(,19000);'
);
function TransArea
Content4() {
Areal.filters.item(0).Apply();
Areal.filters.item(0).Transi
tion = 3;
ArealContent1.style.visibility = 'hidden';
ArealContent4.style.visibility
= 'visible';
document.FlashArealContent4.rewind();
document.FlashArealContent4.play(
);
Areal.filters.item(0).play(1);
window.setTimeout('TransAreaContent5(,
19000);'
);
function TransAreaContent5() {
Areal.filters.item(0).Apply
();
Areal.filters.item(0).Transition = 3;
ArealContent4.style.visibility =
'hidden';
ArealContent5.style.visibility = 'visible';
document.FlashArealContent5
.rewind();
document.FlashArealContent5.play(1);
Areal.filters.item(0).play(1);
window.setTimeout('TransAreaContent6(,19000);'
);
function TransArea
Content6() {

```

```

tion = 2;
ArealContent5.style.visibility = 'hidden';
ArealContent6.style.visibility
= 'visible';
document.FlashArealContent6.rewind();
document.FlashArealContent6.play(
);
Areal.filters.item(0).play(1);
window.setTimeout('TransAreaContent7(,
11000);'
);
function TransAreaContent7() {
Areal.filters.item(0).Apply
();
Areal.filters.item(0).Transition = 3;
ArealContent6.style.visibility =
'hidden';
ArealContent7.style.visibility = 'visible';
document.FlashArealContent7
.rewind();
document.FlashArealContent7.play(1);
Areal.filters.item(0).play(1);
window.setTimeout('TransAreaContent8(,17000);'
);
function TransArea
Content8() {
Areal.filters.item(0).Apply();
Areal.filters.item(0).Transi
tion = 3;
ArealContent7.style.visibility = 'hidden';
ArealContent8.style.visibility
= 'visible';
document.FlashArealContent8.rewind();
document.FlashArealContent8.play(
);
Areal.filters.item(0).play(1);
window.setTimeout('TransAreaContent9(,
11000);'
);
function TransAreaContent9() {
Areal.filters.item(0).Apply
();
Areal.filters.item(0).Transition = 2;
ArealContent8.style.visibility =
'hidden';
ArealContent9.style.visibility = 'visible';
document.FlashArealContent9
.rewind();
document.FlashArealContent9.play(1);
Areal.filters.item(0).play(1);
window.setTimeout('TransAreaContent10(,13000);'
);
function TransArea
Content10() {
Areal.filters.item(0).Apply();
Areal.filters.item(0).Transi
tion = 3;
ArealContent9.style.visibility = 'hidden';
ArealContent10.style.visibility
= 'visible';
document.FlashArealContent10.rewind();
document.FlashArealContent10.pla
y(1);
Areal.filters.item(0).play(1);
window.setTimeout('TransAreaContent11(,
19000);'
);
function TransAreaContent11() {
Areal.filters.item(0).Apply
();
Areal.filters.item(0).Transition = 2;
ArealContent10.style.visibility =
'hidden';
ArealContent11.style.visibility = 'visible';
document.FlashArealContent
11.rewind();
document.FlashArealContent11.play(1);
Areal.filters.item(0).play(1);
window.setTimeout('TransAreaContent1(,17000);'
);
function TransArea

```

```

Content1() {
Areal.filters.item(0).Apply();
Areal.filters.item(0).Transi
tion = 2;
ArealContent22.style.visibility = 'hidden';
ArealContent1.style.visibility
= 'visible';
document.FlashArea3Content1.rewind();
document.FlashArea3Content1.play(
);
Areal.filters.item(0).play(1);
window.setTimeout('TransArea3Content2(,
11000);'
);
function TransArea3Content2() {
Areal.filters.item(0).Apply
();
Areal.filters.item(0).Transition = 2;
ArealContent1.style.visibility =
'hidden';
ArealContent22.style.visibility = 'visible';
Areal.filters.item(0).play
(1);
window.setTimeout('TransArea3Content3(,11000);'
);
function TransArea
3Content3() {
Areal.filters.item(0).Apply();
Areal.filters.item(0).Transi
tion = 3;
ArealContent22.style.visibility = 'hidden';
ArealContent33.style.visibility
= 'visible';
Areal.filters.item(0).play(1);
window.setTimeout('TransArea3Content4(,
11000);'
);
function TransArea3Content4() {
Areal.filters.item(0).Apply
();
Areal.filters.item(0).Transition = 2;
ArealContent33.style.visibility =
'hidden';
ArealContent44.style.visibility = 'visible';
Areal.filters.item(0).play
(1);
window.setTimeout('TransArea3Content5(,11000);'
);
function TransArea
3Content5() {
Areal.filters.item(0).Apply();
Areal.filters.item(0).Transi
tion = 3;
ArealContent44.style.visibility = 'hidden';
ArealContent55.style.visibility
= 'visible';
document.FlashArea3Content5.rewind();
document.FlashArea3Content5.play(
);
Areal.filters.item(0).play(1);
window.setTimeout('TransArea3Content6(,
11000);'
);
function TransArea3Content6() {
Areal.filters.item(0).Apply
();
Areal.filters.item(0).Transition = 2;
ArealContent55.style.visibility =
'hidden';
ArealContent66.style.visibility = 'visible';
Areal.filters.item(0).play
(1);
window.setTimeout('TransArea3Content7(,11000);'
);
function TransArea
3Content7() {
Areal.filters.item(0).Apply();
Areal.filters.item(0).Transi
tion = 3;
ArealContent66.style.visibility = 'hidden';
ArealContent77.style.visibility
= 'visible';

```

```

ty = 'visible';
Areal.filters.item(0).play(1);
window.setTimeout('TransArea3Content8(,
11000);'
);
function TransArea3Content8() {
Areal.filters.item(0).Apply
();
Areal.filters.item(0).Transition = 3;
ArealContent77.style.visibility =
'hidden';
ArealContent88.style.visibility = 'visible';
Areal.filters.item(0).play
(1);
window.setTimeout('TransArea3Content9(,11000);'
);
function TransArea
3Content9() {
Areal.filters.item(0).Apply();
Areal.filters.item(0).Transi
tion = 2;
ArealContent88.style.visibility = 'hidden';
ArealContent99.style.visibility
= 'visible';
document.FlashArea3Content9.rewind();
document.FlashArea3Content9.play(
);
Areal.filters.item(0).play(1);
window.setTimeout('TransArea3Content10(,
19000);'
);
function TransArea3Content10() {
Areal.filters.item(0).Apply
();
Areal.filters.item(0).Transition = 3;
ArealContent99.style.visibility =
'hidden';
ArealContent1010.style.visibility = 'visible';
Areal.filters.item(0).pla
y(1);
window.setTimeout('TransArea3Content11(,11000);'
);
function TransAre
a3Content11() {
Areal.filters.item(0).Apply();
Areal.filters.item(0).Transi
tion = 3;
ArealContent1010.style.visibility = 'hidden';
ArealContent1111.style.visibility
= 'visible';
Areal.filters.item(0).play(1);
window.setTimeout('TransArea3Content12(,11000);'
);
function TransAre
a3Content12() {
Areal.filters.item(0).Apply();
Areal.filters.item(0).Transi
tion = 3;
ArealContent1111.style.visibility = 'hidden';
ArealContent1212.style.visibility
= 'visible';
Areal.filters.item(0).play(1);
window.setTimeout('TransArea3Content13(,11000);'
);
function TransAre
a3Content13() {
Areal.filters.item(0).Apply();
Areal.filters.item(0).Transi
tion = 2;
ArealContent1212.style.visibility = 'hidden';
ArealContent1313.style.visibility
= 'visible';
Areal.filters.item(0).play(1);
window.setTimeout('TransArea3Content14(,11000);'
);
function TransAre
a3Content14() {
Areal.filters.item(0).Apply();
Areal.filters.item(0).Transi
tion = 3;

```

Page 5

Page 6

Page 7

Page 8


```

div> **** <div
id="Area3Content16" style="Position: ABSOLUTE; Top:0%; LEFT
:0%; Width: 100%;
height:100%; color:black; font-family:Arial;f
ont-size:14pt;
visibility:"hidden"; z-index:0;"> 
</div> ****
<div id="Area3Content17" style="Position: ABSOLUTE; Top:0%;
LEFT:0%; Width:
100%; height:100%; color:black;
font-family:Arial;font-size:14pt; visibility:"hidden"; z-i
ndex:0;">
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://active.macromedia.com/flash2/cabs/swf/cab#version=
4,0,0,0"
ID="FlashArea3Content17" WIDTH=255 HEIGHT=651" <PARAM NAME=movie
VALUE="763e412692d3e3d06b81e142f88e76ff2.swf"><PARAM NAME=quality VALUE
=high><PARAM
NAME=mode VALUE=transparent><PARAM NAME=bgcolor VALUE=#FFFFFF><PARAM
NAME=PLAY
VALUE=false><PARAM NAME=LOOP VALUE=false></OBJECT> </
div> **** <div
id="Area3Content18" style="Position: ABSOLUTE; Top:0%; LEFT
:0%; Width: 100%;
height:100%; color:black; font-family:Arial;f
ont-size:14pt;
visibility:"hidden"; z-index:0;"> 
</div> ****
<div id="Area3Content19" style="Position: ABSOLUTE; Top:0%;
LEFT:0%; Width:
100%; height:100%; color:black;
font-family:Arial;font-size:14pt; visibility:"hidden"; z-i
ndex:0;">
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://active.macromedia.com/flash2/cabs/swf/cab#version=
4,0,0,0"
ID="FlashArea3Content19" WIDTH=255 HEIGHT=651" <PARAM NAME=movie
VALUE="17fca4c6e595c2c0450e75fc7d4203.swf"><PARAM NAME=quality VALUE
=high><PARAM
NAME=mode VALUE=transparent><PARAM NAME=bgcolor VALUE=#FFFFFF><PARAM
NAME=PLAY
VALUE=false><PARAM NAME=LOOP VALUE=false></OBJECT> </
div> **** <div
id="Area3Content20" style="Position: ABSOLUTE; Top:0%; LEFT
:0%; Width: 100%;
height:100%; color:black; font-family:Arial;f
ont-size:14pt;
visibility:"hidden"; z-index:0;"> 
</div> ****
<div id="Area3Content21" style="Position: ABSOLUTE; Top:0%;
LEFT:0%; Width:
100%; height:100%; color:black;
font-family:Arial;font-size:14pt; visibility:"hidden"; z-i
ndex:0;">

</div> **** <div id="Area3Content22" style="Position: ABSOL
UTE; Top:0%;
LEFT:0%; Width: 100%; height:100%; color:black;
font-family:Arial;font-size:14pt; visibility:"hidden"; z-i
ndex:0;">

</div> **** </Div> **** </BODY> </HTML>

```

John